

PROJET : CHALLENGE PERSYCU

	NOM PRÉNOM	FONCTION	N° ÉTUDIANT
MAÎTRE D'OUVRAGE	PELLIER DAMIEN	ENSEIGNANT	—
	ASOYAN ARMEN	ÉTUDIANT	11714531
MAÎTRES D'OEUVRE	MACÉ GABRIEL	ÉTUDIANT	12010249
	VELLETAZ CÉDRIC	ÉTUDIANT	11908167
	ZARCH VASSILI	ÉTUDIANT	12107374

Sommaire

Introduction	2
Organisation du projet	2
Pilotage du projet	3
Deadlines	3
Tâches	3
Planification	3
Gestion des risques	4
Livrables	4
Glossaire	5
Index	5

Introduction

Ce document vise à formaliser l'organisation du groupe et son plan de développement dans la réalisation de ce projet.

Organisation du projet

Le groupe est composé de quatre étudiants de troisième année de licence MIAHS.
Le professeur référent de ce projet est monsieur Damien Pellier, il constitue un apport technique et pédagogique.
Les membres du fablab ont aussi eu un apport technique.

Pilotage du projet

Deadlines

La date de fin du projet est le: 28/11/2022

La date de rendu des livrables est le : 5/12/22

Tâches

Une première tâche est la création des attributs, constructeurs et méthodes, ainsi que la réalisation de tests (voir plan de test) des classes des packages moteurs et sensors :

Motors :

- Actionneur : Permet d'interagir avec les moteurs des roues : récupérer les valeurs des tachomètres, avancer ou reculer en ligne droite, effectuer une rotation sur place ou en avançant, récupérer et modifier la vitesse et l'accélération des roues.
- Pince : Permet d'interagir avec le moteur des pinces : ouvrir et fermer les pinces, récupérer et modifier la vitesse du moteur des pinces.

Sensors:

- CameraInfrarouge : Permet d'activer le lien avec la caméra-infrarouge et de récupérer les informations de celle-ci
- Color : Permet d'activer et de récupérer les informations du capteur de couleur.
- Touch : Permet d'activer et de récupérer les informations du capteur de touchée
- Distance : Permet d'activer et de récupérer les informations du capteur ultrasonique.

Une seconde tâche est la création des différents automates des stratégies à mettre en place.

Une autre tâche est la création de la classe agent : C'est la classe qui lie les classes précédentes et qui permet de diriger notre robot. Elle permet de récupérer les premiers palets sans effectuer de recherche, de rechercher un palet, puis de le récupérer et l'amener dans l'objectif adverse, tout en évitant les possibles collisions avec un adversaire ou un mur.

Une dernière tâche sera l'optimisation des stratégies et du code.

Planification

- 1) Découverte du robot et de la librairie leJos ev3 ainsi que sa documentation .
- 2) Création des classes des packages sensors et moteurs : Attributs, constructeurs et méthodes, tout en effectuant des tests de vérification.

- 3) Création d'automates des stratégies (de recherche, récupération, marquer).
- 4) Création de la classe agents et tests
- 5) Optimisation
- 6) Compétition
- 7) Finalisation des livrables

Gestion des risques

Un premier risque est la mésentente au sein du groupe : la solution sera l'explicitation du désaccord.

Un second risque est la détérioration ou la perte du matériel fourni : nous l'utiliserons avec grandes précautions, et le remettrons toujours à la même place après utilisation.

Une source de risque est la création parallèle de code dans une même classe:

- Création de méthodes similaires, ou à partir de méthodes ou attributs modifiés : Pour éviter cela nous récupérerons et mettrons à jour régulièrement le code présent sur le github commun.
- Écrasement de code lors d'une mise à jour du code : récupération régulière du code, et comparaison des différences avant la mise à jour, ou en cas d'erreur, récupérer les éléments supprimés grâce à l'historique de github

Le risque principale étant que le projet ne remplisse pas les critères d'acceptabilité du projet, une première version qui réussit le test d'homologation sera réalisée.

Livrables

Un cahier des charges
Un plan de développement
Un plan de tests
La documentation interne du code + code source
Un rapport de projet

Glossaire

Automates: Dispositif reproduisant en autonomie une séquence d'actions prédéterminées sans l'intervention humaine.

Github: Service web d'hébergement et de gestion de développement de logiciels.

Sensor: Traduction littérale de capteur : Dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable

Actionneur: Objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système.

Références

Notre Github : <https://github.com/InitiationIA/InitiationIA>

<https://lejos.sourceforge.io>

<https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:robotics>

<https://fr.wikipedia.org/wiki/Actionneur>

<https://fr.wikipedia.org/wiki/Automate>

<https://fr.wikipedia.org/wiki/GitHub>

<https://fr.wikipedia.org/wiki/Capteur>

Index

Tâches : page 3

Organisation : page 3

Risque : page 4